IMPROVING LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION BY COMBINING GMM-BASED AND RESERVOIR-BASED ACOUSTIC MODELING

Fabian Triefenbach, Kris Demuynck, Jean-Pierre Martens

Ghent University - IBBT, ELIS Multimedia Lab
Sint-Pietersnieuwstraat 41, B-9000, Ghent, Belgium
fabian.triefenbach@elis.ugent.be

ABSTRACT

In earlier work we have shown that good phoneme recognition is possible with a so-called reservoir, a special type of recurrent neural network. In this paper, different architectures based on Reservoir Computing (RC) for large vocabulary continuous speech recognition are investigated. Besides experiments with HMM hybrids, it is shown that a RC-HMM tandem can achieve the same recognition accuracy as a classical HMM, which is a promising result for such a fairly new paradigm. It is also demonstrated that a state-level combination of the scores of the tandem and the baseline HMM leads to a significant improvement over the baseline. A word error rate reduction of the order of 20% relative is possible.

Index Terms— continuous speech recognition, reservoir computing, tandem acoustic modeling

1. INTRODUCTION

The idea of using an artificial neural network (ANN) for continuous speech recognition has already been discussed in many publications [1, 2, 3, 4, 5] and good results were obtained with various network types, including feed-forward neural networks (multi-layer perceptrons), recurrent neural networks and Restricted Boltzmann Machines which are stacked on top of each other to form so-called deep belief networks [6, 7, 8]. The posterior probabilities computed by such networks can either be used in an HMM hybrid [1] where the scaled network outputs constitute the state likelihoods, or in a ANN-HMM tandem [2], where the network outputs are used as input features of an otherwise conventional HMM system. The hybrid approach can either work with a restricted set of acoustic classes representing monophones or with a much larger set of acoustic classes representing triphones.

Here we will examine a special kind of neural networks, called reservoirs, and computation with such networks is generally called Reservoir Computing [9]. A reservoir is a pool of non-trained recurrently connected non-linear computational nodes. The reservoir is combined with a set of linear nodes that "read out" the reservoir state, defined as the values at the outputs of the reservoir nodes. The linear nodes are therefore often referred to as readout nodes and they are trained to represent acoustic classes, monophones in our case [10]. Reservoir Computing combines the capacity of a recurrent neural network to model long-term signal dynamics with the advantage of just having to solve a set of linear equations to find the optimal weights. In other words, it circumvents the critical back-propagation-through-time (BPTT) training procedure that is normally needed for training conventional recurrent neural networks [11].

RC has already been applied successfully to speech recognition, reaching fairly good results for phoneme recognition [10, 12] and continuous digit recognition in a noisy environment [13]. In what follows we briefly review the concepts of RC, we introduce different RC-HMM system architectures for large vocabulary continuous speech recognition (LVCSR) and we present experimental results showing that by means of model combination a significant improvement over a state-of-the-art HMM baseline can be attained.

2. RESERVOIR COMPUTING

The reservoirs in our systems are Echo State Networks [9] with a linear readout layer (see Figure 1). The reservoir input weights and the weights of the recurrent connections are first drawn from a normalized random distribution and then properly rescaled to assert a stable dynamical system with a good balance between excitability by new inputs and by past inputs, represented in the reservoir state (for details see [9, 10]). The reservoir computes the state vector $\mathbf{x}[t]$ at time t from the previous state vector and the current input $\mathbf{u}[t]$ as

$$\mathbf{x}[t] = f_{res}(\mathbf{W}_{res}\mathbf{x}[t-1] + \mathbf{W}_{in}\mathbf{u}[t]) \qquad (1)$$

The weight matrices $\mathbf{W}_{res}$ and $\mathbf{W}_{in}$ represent the weights of the interconnections and f_{res} represents the non-linearity applied to the node activation. In the present work this non-linearity is a $tanh()$. The reservoir state is supplied to a layer of linear units, called the readout nodes, or even more briefly, the readouts. The weights of the connections from the reservoir nodes to the readouts are trained to constitute the best linear regression of the monophone classes.

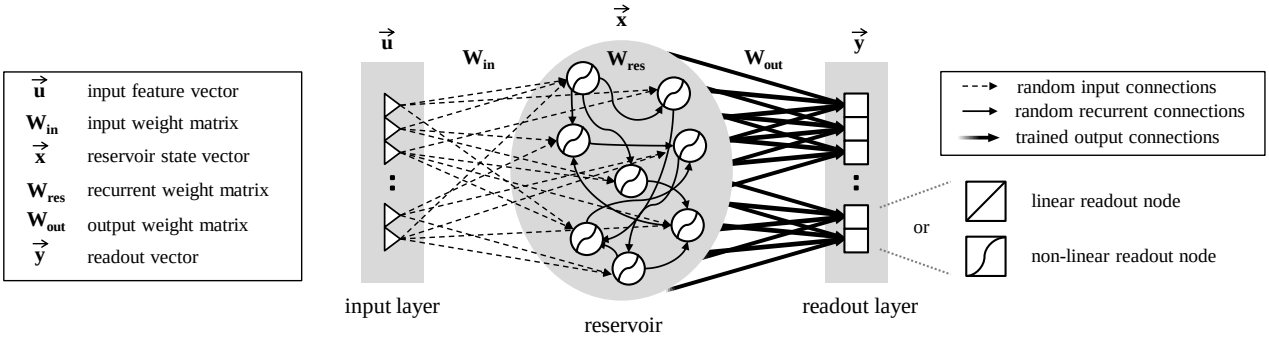

Fig. 1. *Reservoir Computing network with randomly-fixed input weights $\mathbf{W}_{in}$, randomly-fixed recurrent connections $\mathbf{W}_{res}$, and trainable readout weights $\mathbf{W}_{out}$.*

The optimal weight matrix $\mathbf{W_{out}}$ is found by minimizing the mean-squared error (MSE). If the reservoir states of the training data form the N_t rows of a matrix $\mathbf{X}$ and if the corresponding desired training targets form the rows of a matrix $\mathbf{D}$, then $\mathbf{W_{out}}$ is given by

$$\mathbf{W}_{out} = \arg\min_{\mathbf{W}} \left(\frac{1}{N_t} \left(||\mathbf{X}\,\mathbf{W} - \mathbf{D}||^2 \right) + \epsilon \; ||\mathbf{W}||^2 \right) \quad (2)$$

with ϵ being a regularization term. The closed-form solution of the minimization problem is given by

$$\mathbf{W}_{out} = (\mathbf{X^T X} + \epsilon\,\mathbf{I})^{-1}(\mathbf{X^T D}) \quad (3)$$

with $\mathbf{I}$ being the unity matrix. Note that replacing the linear readout units by sigmoidal units can be helpful to improve the performance [12], but it is not considered in the present work because it calls for a more time-consuming stochastic gradient-descent training method. Such a GD-based training does become interesting however in the case of a very big reservoir because in that case the inversion of a very large matrix $\mathbf{X^T X}$ becomes problematic.

3. RESERVOIR-BASED SYSTEMS

In this section we introduce our HMM baseline system and we propose the different reservoir-based architectures that will be experimentally evaluated later.

3.1. The HMM baseline

The HMM baseline is developed using the SPRAAK toolkit[1]. It comprises triphone states that emerged from a decision tree based clustering, and each state is modeled by a GMM. All GMMs select elements from a global pool of Gaussians. By definition, such an approach implies extensive parameter tying. All triphones are modeled by a 3-state left-to-right HMM. The model complexity (number of Gaussians, states, mixtures) is controlled by the training process and therefore dependents on the properties of the available training data.

3.2. The RC-HMM hybrid

In a previous publication [10] we already elaborated a RC-HMM hybrid according to the principles outlined in [1]: the context-independent readouts are converted to acoustic state likelihoods by Bayes' law and a Viterbi-search is used to find the most likely path through an HMM which models the utterance structure. However, the hybrid in [10] was designed for phoneme recognition with a bigram phonotactic utterance model. In the present work, a word-level language model and a pronunciation dictionary are introduced to construct a hybrid for large vocabulary continuous speech recognition (LVCSR).

There are reasons to believe though that the sketched approach to LVCSR is sub-optimal. In fact, for phoneme recognition it suffices that the pronounced phoneme is – in the right time interval – the winning hypothesis generated by the reservoir. For LVCSR on the other hand, there is a realistic chance that the pronunciation of the correct word sequence, as it emerges from the pronunciation dictionary, differs from the actually spoken phoneme sequence. The desired word sequence can only pop up as the solution then if the likelihoods of its constituent phonemes are sufficiently high. This implies that the likelihoods of competing hypotheses are of great importance, too. A classical solution for handling such pronunciation mismatches is to introduce context-dependent phonemes. If a phoneme is frequently articulated as another sound in a specific phonemic context then the training examples of that context-dependent phoneme will mainly consist of utterances of this other sound. As such, the phoneme emerging from the phonetic dictionary will get a sufficiently large likelihood, even though strictly speaking, it has not been pronounced. Consequently, we expect that a hybrid with monophone classes may not yield the best LVCSR performance. This assumption is actually supported by recently published work on deep neural networks where

[1] SPRAAK: Speech Processing, Recognition and Automatic Annotation Kit [http://www.spraak.org]

systems employing monophone classes were outperformed by similar systems employing context-dependent (CD) phonetic classes [5, 4]. Building CD-RC-HMM hybrids will not be investigated here because we first want to establish how well CI-RC-HMM hybrids perform. However, we expect that moving to context-dependent systems constitutes an important direction for future work.

3.3. The RC-HMM tandem

An RC-HMM tandem is a standard context-dependent HMM comprising GMM modeling in terms of the reservoir readouts as the acoustic features. Figure 2 shows the complete tandem architecture. The upper part of the figure represents the reservoir-based feature extractor. The bottom part shows the conventional GMM-based context-dependent HMM system architecture, including two pre-processing steps that normalize and decorrelate the readouts of the reservoir respectively.

As in other published ANN-HMM tandems we introduce a first pre-processing step which consists of a non-linear transformation. The aim is to reduce the skewness of the feature distributions so as to make them more suitable for Gaussian modeling. In the case of an MLP one either takes the logarithm of each output or select the so-called activation of the output nodes [2, 14]. In the case of RC with linear regression we argue that there is no need to do this because the readouts are linear functions of the reservoir state variables and therefore they should not exhibit a skewed distribution. This argumentation was in fact confirmed by experimental evidence. The pre-processing step only needs to be applied when logistic regression was used to train the readout weights.

The second pre-processing step decorrelates the features in order to facilitate their modeling by GMMs with diagonal covariance matrices [2]. It is achieved by means of a linear feature transformation that can also be utilized to reduce the dimensionality of the vector [14] at the same time. In the literature different methods for accomplishing this have been proposed. In the present work we apply Mutual Information Discriminative Analysis (MIDA) [15], a technique that can be viewed as a special form of Linear Discriminant Analysis (LDA).

We conjecture that dimensionality reduction will be inevitable in case we create an RC-HMM tandem that, in analogy with [16], supplies both the original MFCC vectors and the reservoir readouts to the HMM.

3.4. Taking phonetic confusions into account

Analyzing the readouts of a trained reservoir system revealed that the frame-wise winner not always corresponds to the desired target, defined as the target that would emerge from an alignment of the speech with its transcription, derived from the orthography of the speech and the pronunciation dictionary of the recognizer. These mismatches can be captured in a phonetic confusion matrix which is retrieved from alignments of the reservoir readouts and the dictionary-based transcriptions of a sufficiently large set of development utterances. We will use this confusion matrix to manipulate the readouts before supplying them to our hybrid and tandem systems.

3.5. Model combination

Having two different acoustic models based on different paradigms at our disposal, it is possible to investigate whether these two systems make the same or different errors. As we observed some complementarity between the two models, we have also investigated the combination of a RC-HMM-tandem and a conventional CD-HMM.

Both individual systems employ the same acoustic units (triphones) with the same topology, but different methods to assess the state likelihoods. Consequently, a simple way to combine the two is to combine their acoustic likelihoods at the state level. Where a (weighted) linear combination of likelihoods is ideal for reducing modeling noise, a (weighted) log-linear combination is preferred for combining complementary information streams because it complies better with the log-linear combination of likelihoods across frames. For simplicity, we only worked with a single (state-independent) weight that is determined from a recognition experiment on the development data. Such a state-based combination scheme assumes that the composing acoustic models are time synchronous, i.e. state transitions in each model occur simultaneous. This simplifies the decoding and is a reasonable assumption given that all acoustic models in our experiments were bootstrapped from the same initial state-level segmentation of the training data.

There are of course practical issues to attend. For instance, if decision tree clustering of triphone states is applied in the individual systems, the emerging decision trees will almost certainly differ. This calls for the composition of a larger (deeper) decision tree with leaf nodes corresponding to unique combinations of states from the individual systems. The emission log-likelihoods of the combined states are calculated as a weighted sum of the log-likelihoods emerging from the emission distributions of the composing states. For the transition probabilities, a weighted average of the transition probabilities of the composing states is used (i.e. a linear instead of log-linear combination). The state-dependent weights used for this averaging are chosen proportional to the cumulative state posteriors as recorded in the last training pass of the individual systems.

4. EXPERIMENTAL RESULTS

4.1. Experimental conditions

In what follows we perform LVCSR experiments on TIMIT [17] and WSJ0 [18]. The TIMIT database is normally designed to perform phoneme recognition experiments but it has

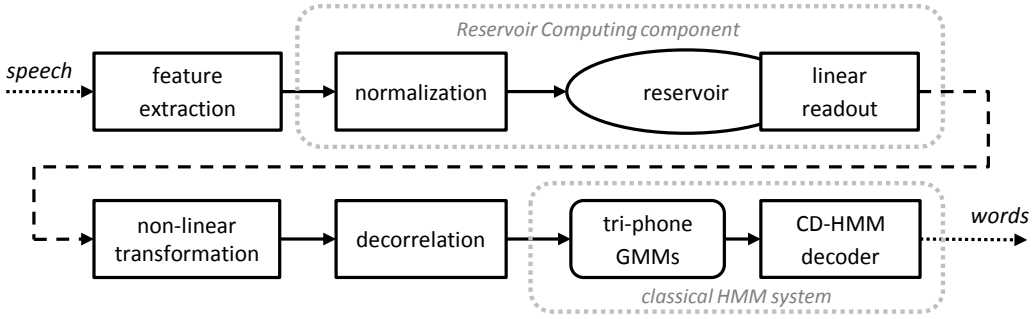

Fig. 2. *The reservoir-HMM tandem architecture: (1) the feature extractor, the Reservoir Computing component, the post-processing of the readouts and the GMM-based CD-HMM decoder.*

also been used for continuous speech recognition experiments (see [19] for details). In all experiments the raw acoustic features are the log energy, $c_1 \dots c_{12}$ (MFCCs) and the Δ's and $\Delta\Delta$'s thereof. Since we did already observe good phone recognition accuracy using these features [10], there was no reason for not using them for LVCSR as well. The preprocessing of the HMM systems also embeds utterance-based mean-normalization of the static features before computing the dynamic features.

For experiments on TIMIT we employ a lexicon containing all 6100 words appearing in the TIMIT sentences (SX and SI sentences only). The word pronunciations as derived from the CMU pronunciation dictionary[2]. The language model is a back-off bigram derived from all TIMIT utterances. It has a perplexity of 89.3 (see [19]). We held out 48 training speakers to compose a development set and we report word error rates (WERs) on the complete test set, comprising 9454 word tokens.

For experiments on WSJ0 we employ the 5K lexicon distributed with the corpus. The pronunciations were also taken from the CMU dictionary. For the speech data, we discern between the standard training set (84 speakers), the *dev92* development set (10 speakers) and the *nov92* evaluation set (8 speakers). For development and testing we only use the utterances without verbal punctuation. The language model is the trigram LM for the 5k closed vocabulary task (delivered with the corpus). We report WERs on the evaluation set.

The reservoirs involved in our experiments are constructed as outlined in [10]. Since we formerly achieved good phoneme recognition on TIMIT with a reservoir of 20K nodes, we sticked to that reservoir size for the LVCSR experiments on TIMIT as well. The reservoir is combined with a readout layer representing 40 phonetic classes used in the CMU dictionary (excluding stress information). For WSJ0 we noticed no significant differences between the phonetic classification on the training and development data for a reservoir of that size, suggesting that larger reservoirs can be used here. For the time being, we have worked with a reservoir of 30K because this is the largest size for which we can compute the closed-form solution on a normal workstation. The reservoir is again combined with a readout layer that represents 40 phonetic classes.

[2]The Carnegie Mellon University Pronouncing Dictionary [http://www.speech.cs.cmu.edu/cgi-bin/cmudict]

4.2. Baseline HMM-systems

The acoustic models of the TIMIT baseline share a pool of 5500 Gaussians, they comprise 500 tied states and 1200 cross-word triphones. On average, the GMMs are composed of 139 elements selected from the pool of Gaussians. The acoustic models of the WSJ0 baseline share a pool of 17500 Gaussians, they comprise 1000 tied states and 4400 cross-word triphones. On average, the GMMs are composed of 172 elements selected from the pool of Gaussians. The number of Gaussians, states, triphones and mixtures are determined automatically from the size and the statistics of the training data, so as to keep the risk of over-fitting low. The performances of our baseline systems are listed in Table 1. It can be verified that we have an excellent baseline: compare to [19] for TIMIT and to [3] for WSJ0.

4.3. The RC-HMM hybrids

We only constructed the RC-HMM hybrid for TIMIT. The reservoir training was performed using state-wise targets that were retrieved from a segmentation created with our baseline HMM. A first system, relying on the assumption that zero-clipped readouts can be linearly transformed to posterior probabilities, achieved a WER as large as 8.5%. By applying a trained non-parametric mapping of readouts to posterior probabilities (as shown in [13]), we could reduce this WER to 7.0%. By introducing a confusion matrix learned on the development data, we could further reduce the WER to 6.1%. However, this WER is still far above the 3.7% that we achieved with our HMM baseline.

Although the results of the context-independent RC-HMM hybrid were not satisfying, we wanted to establish whether this is due to the fact that context-independent pho-

Table 1. *WER (in %) on TIMIT and WSJ0 test sets.*

architecture	TIMIT	WSJ0
BASELINE		
CD-HMM	3.7	3.8
RESERVOIR-TANDEMS		
Tandem A (only readouts)	3.8	4.8
Tandem B (readouts + MFCCs)	3.4	4.0
LIKELIHOOD MERGING		
Tandem A + CD-HMM	2.8	3.3
Tandem B + CD-HMM	2.7	3.4

netic classes were used, or to the fact that the reservoir readouts do not carry all the necessary information about the speech anymore. To that end, we investigated RC-HMM tandems where the readouts are simply treated as acoustic observations. If competitive results could be obtained in this way, this would then be interpreted as proof that that the readouts are sufficiently informative but that the method of exploiting this information in the RC-HMM hybrid is suboptimal.

4.4. The RC-HMM tandems

We investigated two tandem configurations: *Tandem A* employs the 40 reservoir readouts as the acoustic observations while *Tandem B* employs them in combination with the original 39 MFCCs. In the latter tandem, the dimensionality of the feature vector is reduced to 39 by means of MIDA. In both cases, the phonetic confusion matrix was applied to the readouts. The results listed in Table 1 show that *Tandem A* closely approaches the baseline results on TIMIT, but not on WSJ0. We conjecture that the data show that the reservoir readouts are sufficiently informative, provided that the reservoir is large enough. We believe that experiments with even larger reservoirs will eventually confirm this for the WSJ0 case too.

The figures clearly show that *Tandem B* outperforms *Tandem A*. However, since *Tandem B* (MFCCs + readouts) does not significantly outperform the HMM baseline (MFCCs alone) one might come to the conclusion that the temporal processing that gave rise to the reservoir readouts does not add valuable information complementary to the information retrieved directly from the MFCCs.

4.5. The combinations of models

As opposed to the previous conclusion, we observed that the decision trees of the HMM baseline and *Tandem A* do show a lot of overlap, but nevertheless, they are different to some extend and they result in different tied states. This observation suggests that the reservoir features and the MFCCs are somewhat complementary in their phonetic specializations. In order to investigate this complementarity in more detail we compared the recognition outputs of the two systems on the development data. This comparison revealed that in 160 of the 794 processed sentences, these outputs were different. We also observed that in situations where only minor errors occur (e.g. a single word deletion/insertion/substitution), usually one of the two systems finds the correct hypotheses (110 cases). Even though we could not identify any systematic pattern in the recognition differences, we contemplate that the above analysis provides sufficient support for the hypothesis that the readouts and the MFCC features do carry complementary information, but that their early fusion in the feature stream is not capable of exploiting that information adequately. That is why we investigated late fusion at the level of the states as an alternative. This late fusion boils down to combining the state likelihoods of the individual tandems as explained before.

The merging results listed in Table 1 clearly show that the combination of *Tandem A* with the baseline CD-HMM leads to a significant improvement for both tasks. The relative improvements are no less than 24% for TIMIT and 13% for WSJ0. For the sake of completeness, and because *Tandem B* outperformed *Tandem A*, we also tested the merging the CD-HMM baseline and *Tandem B*. This merging of models with partly overlapping inputs did not outperform the merging with independent inputs (CD-HMM and *Tandem A*).

Obviously, adding the RC component needed for the tandem adds complexity to the system. However, measurements on the TIMIT system showed that the total recognition time is only increased by a factor 1.3, which is acceptable given the significant improvement being obtained, and given the fact this factor could be further reduced by a more rigorous exploitation of connection sparsity and weight sharing inside the reservoir. Remarkable is that the combined system has more tied states but that this does not result in a higher computational load because the better acoustic models enable a more efficient pruning of hypotheses.

5. CONCLUSIONS & FUTURE WORK

In this paper we showed that combining a Reservoir Computing based HMM tandem with a conventional GMM-HMM system leads to improvements of large vocabulary continuous speech recognition. Substantial gains were observed on both TIMIT (24% relative) and WSJ0 (13% relative), and there are reasons to believe that the gain on WSJ can be further raised by employing a larger reservoir. The reason why we did not do this yet is because for these larger sizes, we either have to use a gradient-descent training of he reservoir, or we have to stack reservoirs on top of each other (as we did in [10] for phoneme recognition).

We believe to have demonstrated that the fairly new paradigm of Reservoir Computing has potential to further improve state-of-the-art speech recognition in the near future.

The temporal processing in the reservoir seems capable of producing information at time t that is not available in the MFCC vector at time t and that is not retrievable by a simple Markov modeling approach. This also motivates us to continue our efforts to narrow the performance gap between a RC-HMM hybrid and a state-of-the-art HMM, e.g. by exploring approaches to handle pronunciation mismatches and by exploring reservoirs with readouts that represent context-dependent phonetic classes.

6. ACKNOWLEDGMENTS

The work presented in this paper was funded by the EC FP7 project ORGANIC (FP7-231267) and by the RECAP project funded by the Research Foundation Flanders (FWO-Vlaanderen).

7. REFERENCES

[1] Herve A. Bourlard and Nelson Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[2] H. Hermansky, D.P.W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional hmm systems," in *Proc. of ICASSP*, 2000, pp. 1635 –1638.

[3] Sabato Marco Siniscalchi, Torbjørn Svendsen, and Chin-Hui Lee, "A bottom-up stepwise knowledge-integration approach to large vocabulary continuous speech recognition using weighted finite state machines," in *Proc. of INTERSPEECH*, 2011, pp. 901–904.

[4] Frank Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. of INTERSPEECH*, 2011, pp. 437–440.

[5] G.E. Dahl, Dong Yu, Li Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[6] Li Deng and Dong Yu, "Deep convex net: A scalable architecture for speech pattern classification," in *Proc. of INTERSPEECH*, 2011, pp. 2285–2288.

[7] Li Deng, Dong Yu, and John Platt, "Scalable stacking and learning for building deep architectures," in *Proc. of ICASSP*, 2012, pp. 2133–2136.

[8] Brian Hutchinson, Li Deng, and Dong Yu, "A deep architecture with bilinear modeling of hidden representations: applications to phonetic recognition," in *Proc. of ICASSP*, 2012, pp. 4805–4808.

[9] Herbert Jaeger, "Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the echo state network approach," Tech. Rep., German National Research Center for Information Technology, 2002.

[10] Fabian Triefenbach, Azarakhsh Jalalvand, Benjamin Schrauwen, and Jean-Pierre Martens, "Phoneme recognition with large hierarchical reservoirs," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 2307–2315.

[11] P.J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550 –1560, oct 1990.

[12] Fabian Triefenbach and Jean-Pierre Martens, "Can non-linear readout nodes enhance the performance of reservoir-based speech recognizers?," in *Proc. of IEEE Conference on Informatics & Computational Intelligence*, 2011, pp. 262–267.

[13] Azarakhsh Jalalvand, Fabian Triefenbach, and Jean-Pierre Martens, "Continuous digit recognition in noise: Reservoirs can do an excellent job," in *Proc. of INTERSPEECH*, 2012.

[14] Qifeng Zhu, Barry Chen, Nelson Morgan, and Andreas Stolcke, "On using mlp features in lvcsr," in *Proc. of INTERSPEECH*, 2004, pp. 921–924.

[15] Kris Demuynck, Jacques Duchateau, and Dirk Van Compernolle, "Optimal feature sub-space selection based on discriminant analysis," in *Proc. of EUROSPEECH*, 1999, pp. 1311–1314.

[16] Fabio Valente, Mathew Magimai-Doss, and Wen Wang, "Analysis and comparison of recent mlp features for lvcsr systems," in *Proc. of INTERSPEECH*, 2011, pp. 1245–1248.

[17] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren, "The DARPA TIMIT acoustic-phonetic continuous speech corpus cd-rom," Tech. Rep., NIST, 1993.

[18] Douglas B. Paul and Janet M. Baker, "The design for the wall street journal-based csr corpus," in *Proc. of the workshop on Speech and Natural Language*, 1992, pp. 357–362.

[19] Cheng Yang, Jean-Pierre Martens, Pol Ghesquiere, and Dirk Van Compernolle, "Pronunciation Variation Modeling for ASR: Large Improvements are possible but small ones are likely," in *Proc. of ITRW on PMLA*, 2002, pp. 123–128.